

Proyecto de investigación-Interrupciones

Juan Daniel Gonzalez Puerta

Junio 2020

¿Qué es una interrupción en el contexto de los microprocesadores?

Interrupción (también conocida como corrupción del hardware o petición de interrupción) es una señal recibida por el procesador de un ordenador, indicando que debe "interrumpir" el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación. Si se acepta la solicitud, el procesador responde suspendiendo sus actividades actuales, guardando su estado y ejecutando una función llamada manejador de interrupciones para ocuparse del evento. Una interrupción supone la ejecución temporaria de un programa, para pasar a ejecutar una "subrutina deservicio de interrupción", que pertenece al BIOS.

En otras palabras es una respuesta del procesador a un evento que necesita atención del software. Una condición de interrupción alerta al procesador y sirve como una solicitud para que el procesador interrumpa el código que se está ejecutando actualmente cuando está permitido, de modo que el evento se pueda procesar de forma oportuna.

¿Se puede hablar de la historia de las interrupciones?

Las interrupciones surgen de las necesidades que tienen los dispositivos periféricos de enviar información al procesador principal de un sistema de computación.

El ordenador UNIVAC 1103 es generalmente acreditado con el uso más temprano de interrupciones en 1953. Anteriormente, en el UNIVAC I (1951) "El desbordamiento aritmético activó la ejecución de una rutina de arreglo de dos instrucciones en la dirección 0, o, a elección del programador, hizo que el ordenador se detuviera." El IBM 650 (1954) incorporó la primera aparición de enmascaramiento de interrupción. La Oficina Nacional de Normas DYSEAC (1954) fue la primera en usar interrupciones para E/S. El IBM 704 fue el primero en utilizar interrupciones para depuración, con un "desvío de transferencia", que

podría invocar una rutina especial cuando se encontró una instrucción de bifurcación. El sistema MIT Lincoln Laboratory TX-2 (1957) fue el primero en proporcionar múltiples niveles de interrupciones prioritarias.

¿Qué tipo de interrupciones existen?

Existen tres tipos de interrupciones, las cuales son:

- Interrupciones de hardware: Son interrupciones que se producen como resultado de, normalmente, una operación de entrada/salida. No son producidas por ninguna instrucción sino que son señales que producen los dispositivos externos para indicarle al procesador que necesitan ser atendidos.
- Trampas: Es un tipo de interrupción sincrónica típicamente causada por una condición de error, por ejemplo, una división por 0 o un acceso inválido a memoria en un proceso de usuario. Normalmente genera un cambio de contexto a modo supervisor para que el sistema operativo atienda el error. De manera que podemos ver como las excepciones son un mecanismo de protección que permite garantizar la integridad de los datos.
- Interrupciones de software: A menudo se tiende a confundir las interrupciones de software y las trampas, ya que su naturaleza es bastante similar. Sin embargo las excepciones o interrupciones por software se producen al realizar una operación no permitida por lo que de algún modo podemos decir que no es controlada directamente por el programador sino que, por un fallo al programar, se producen o directamente vinieron programadas después de ejecutarse ciertas instrucciones. No obstante las trampas sí que son provocadas por el programador.

¿Cómo se hace la implementación de interrupciones a nivel de hardware?

El procesador espera a que los dispositivos periféricos le indiquen que ocurrió algún evento mientras este realiza otras actividades, siendo así mucho más eficiente.

Las interrupciones de hardware pueden llegar asincrónicamente con respecto al reloj del procesador, y en cualquier momento durante la ejecución de la instrucción. Por consiguiente, todas las señales de interrupción del hardware se condicionan sincronizándolas con el reloj del procesador, y se actúa sobre ellas sólo en los límites de la ejecución de la instrucción.

¿Cómo se implementan las interrupciones por software?

En general, actúan de la siguiente manera: Un programa en ejecución llega a una instrucción que requiere del sistema operativo para alguna tarea, por ejemplo para leer un archivo en el disco duro (cuando un programa necesita un dato exterior, se detiene y pasa a cumplir con las tareas de recoger ese dato). En ese momento por tanto llama al sistema y se interrumpe virtualmente hasta recibir respuesta, en el ejemplo anterior hasta que no se haya leído el disco y el archivo esté en memoria principal. Durante esa espera las instrucciones que se ejecutarán no serán del programa, sino del sistema operativo. Una vez éste termine su rutina ordenará reanudar la ejecución del programa auto interrumpido en espera. Por último la ejecución del programa se reanuda.

Una interrupción de software es causada intencionadamente por la ejecución de una instrucción especial que, por diseño, invoca una interrupción cuando se ejecuta esto nos indica que realmente no es cuestión del lenguaje de programación si no del diseño previo del software y de las interrupciones que se consideren necesarias después de ciertas instrucciones. Esas instrucciones funcionan de manera similar a las llamadas de subrutinas y se utilizan para diversos fines, como solicitar servicios del sistema operativo e interactuar con los controladores de dispositivos (por ejemplo, para leer o escribir medios de almacenamiento) como son interrupciones controladas netamente por instrucciones realizadas por el procesador estas mismas no se ven afectadas por el uso de un hardware diferente.

Implementación

La imagen “montaje interrupciones” muestra el montaje en Arduino en la plataforma TinKerCad, donde se conecta un sensor de temperatura al Arduino al pin analógico 0 por el pin de salida de voltaje del sensor, se alimenta este sensor con 5 voltios y aparte se conecta un led al pin 11 digital.

Primero inicializamos dos variables las cuales son sensor y umbral, la variable sensor será el número del pin analógico que usaremos y la variable umbral será el umbral de temperatura que elijamos a conveniencia, yo he elegido 25 ya que pretendo “simular” la temperatura que mantendría un procesador sencillo. En el ciclo loop creo una variable llamada lectura la cual me leerá el voltaje del sensor y luego convertiré este valor de voltaje a temperatura guardándolo en una variable llamada temp. Finalmente coloco condicionales que activaran el led una vez la temperatura sobrepase el umbral y de no ser así lo dejara apagado y podrá realizar otras tareas.

Bibliografia

- [1] Rosenthal, Scott (mayo de 1995). "Básicos de Las Interrupciones".
- [2] Colaboradores de Wikipedia. (2020, 2 de junio). Interrumpir. En Wikipedia, La Enciclopedia Libre. desde <https://en.wikipedia.org/w/index.php?title=Interrupt&oldid=960304240>
- [3] Interrupt List, part 4 of 5. www2.informatik.uni-halle.de. Archivado desde el original el 2 de septiembre de 2013.
- [4] Stallings, W., González del Alba Baraja, Á., Joyanes Aguilar, L., Oñate Gómez, A., & Torres Franco, E. (2002). Sistemas operativos (capítulo 1.4). Madrid: Prentice Hall.
- [5] <https://www.fing.edu.uy/tecnoinf/mvd/cursos/arqcomp/material/teo/arq-teo08.pdf>. Arquitectura de Computadoras { Interrupciones. (Basadas en las Notas de Teórico Versión 1.0 del Dpto. de Arquitectura-InCo-FIng)
- [6] Codd, Edgar F. "Multiprogramming". *Advances in Computers*. 3: 82.